

FFFFFFFFFFFFFFFF	111	111	AAAAAAAAA	
FFFFFFFFFFFFFFFF	111	111	AAAAAAAAA	
FFFFFFFFFFFFFFFF	111	111	AAAAAAAAA	
FFF	111111	111111	AAA	AAA
FFF	111111	111111	AAA	AAA
FFF	111111	111111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFFFFFFFFFFFFF	111	111	AAA	AAA
FFFFFFFFFFFFFF	111	111	AAA	AAA
FFFFFFFFFFFFFF	111	111	AAA	AAA
FFF	111	111	AAAAAAAAAAAAAAAA	
FFF	111	111	AAAAAAAAAAAAAAAA	
FFF	111	111	AAAAAAAAAAAAAAAA	
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111	111	AAA	AAA
FFF	111111111	111111111	AAA	AAA
FFF	111111111	111111111	AAA	AAA
FFF	111111111	111111111	AAA	AAA

```
MM      MM      AAAAAA      KK      KK      AAAAAA      CCCCCCCC      CCCCCCCC
MM      MM      AAAAAA      KK      KK      AAAAAA      CCCCCCCC      CCCCCCCC
MMM     MMM     AA      AA      KK      KK      AA      AA      CC      CC
MMM     MMM     AA      AA      KK      KK      AA      AA      CC      CC
MM      MM      AA      AA      KK      KK      AA      AA      CC      CC
MM      MM      AA      AA      KK      KK      AA      AA      CC      CC
MM      MM      AA      AA      KKKKKK      AA      AA      CC      CC
MM      MM      AA      AA      KKKKKK      AA      AA      CC      CC
MM      MM      AAAAAAAAAA      KK      KK      AAAAAAAAAA      CC      CC
MM      MM      AAAAAAAAAA      KK      KK      AAAAAAAAAA      CC      CC
MM      MM      AA      AA      KK      KK      AA      AA      CC      CC
MM      MM      AA      AA      KK      KK      AA      AA      CC      CC
MM      MM      AA      AA      KK      KK      AA      AA      CC      CC
MM      MM      AA      AA      KK      KK      AA      AA      CCCCCCCC      CCCCCCCC
MM      MM      AA      AA      KK      KK      AA      AA      CCCCCCCC      CCCCCCCC
MM      MM      AA      AA      KK      KK      AA      AA      CCCCCCCC      CCCCCCCC
MM      MM      AA      AA      KK      KK      AA      AA      CCCCCCCC      CCCCCCCC
```

```
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
```

```
1 0001 0 MODULE MAKACC (
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000'
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 ++
32 0032 1
33 0033 1 FACILITY: F11ACP Structure Level 1
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This routine makes the necessary hookups in the I/O database to
38 0038 1 reflect a new file access.
39 0039 1 ENVIRONMENT:
40 0040 1
41 0041 1 STARLET operating system, including privileged system services
42 0042 1 and internal exec routines. This routine must be called
43 0043 1 in kernel mode.
44 0044 1
45 0045 1 --
46 0046 1
47 0047 1
48 0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 20-Dec-1976 17:28
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1 V02-001 LMP0005 L. Mark Pilant, 29-Dec-1981 15:05
53 0053 1 Add support for Cathedral windows.
54 0054 1
55 0055 1 A0100 ACG0001 Andrew C. Goldstein, 10-Oct-1978 20:01
56 0056 1 Previous revision history moved to F11A.REV
57 0057 1
```

MAKACC  
V04-000

H 1  
16-Sep-1984 01:09:33  
14-Sep-1984 12:29:42

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[F11A.SRC]MAKACC.B32;1 Page (1)

:	58	0058	1	! **
:	59	0059	1	
:	60	0060	1	
:	61	0061	1	LIBRARY 'SYSS\$LIBRARY:LIB.L32';
:	62	0062	1	REQUIRE 'SRC\$:FCPDEF.B32';

```

64 0377 1 GLOBAL ROUTINE MAKE_ACCESS (FCB, WINDOW, ABD) : NOVALUE =
65 0378 1
66 0379 1 ++
67 0380 1
68 0381 1 FUNCTIONAL DESCRIPTION:
69 0382 1
70 0383 1 This routine makes the necessary hookups in the I/O database to
71 0384 1 reflect a new file access.
72 0385 1
73 0386 1 CALLING SEQUENCE:
74 0387 1 MAKE_ACCESS (ARG1, ARG2, ARG3)
75 0388 1
76 0389 1 INPUT PARAMETERS:
77 0390 1 ARG1: address of FCB to access
78 0391 1 ARG2: address of window to link up
79 0392 1 ARG3: address of buffer descriptors
80 0393 1
81 0394 1 IMPLICIT INPUTS:
82 0395 1 CURRENT_VCB: VCB of volume in process
83 0396 1
84 0397 1 OUTPUT PARAMETERS:
85 0398 1 NONE
86 0399 1
87 0400 1 IMPLICIT OUTPUTS:
88 0401 1 NONE
89 0402 1
90 0403 1 ROUTINE VALUE:
91 0404 1 NONE
92 0405 1
93 0406 1 SIDE EFFECTS:
94 0407 1 VCB transaction count bumped, access counts in FCB adjusted,
95 0408 1 FCB and window hooked up.
96 0409 1
97 0410 1 --
98 0411 1
99 0412 2 BEGIN
100 0413 2
101 0414 2 MAP
102 0415 2 FCB : REF BBLOCK, ! FCB arg
103 0416 2 WINDOW : REF BBLOCK, ! window arg
104 0417 2 ABD : REF BBLOCKVECTOR [ABD$C_LENGTH];
105 0418 2 ! buffer descriptor arg
106 0419 2
107 0420 2 LOCAL
108 0421 2 WINDOW_SEGMENT : REF BBLOCK; ! address of the current window segment
109 0422 2
110 0423 2 EXTERNAL
111 0424 2 CLEANUP_FLAGS : BITVECTOR, ! cleanup action flags
112 0425 2 CURRENT_VCB : REF BBLOCK, ! VCB
113 0426 2 PM$SGL_OPEN : ADDRESSING_MODE (ABSOLUTE),
114 0427 2 ! system count of currently open files
115 0428 2 PM$SGL_OPENS : ADDRESSING_MODE (ABSOLUTE);
116 0429 2 ! system count of files opened
117 0430 2
118 0431 2 ! If the access count in the FCB is zero, hook it into the FCB list,
119 0432 2 ! since it is not there yet. If, however, the directory LRU
120 0433 2 ! bit is set, the FCB is already in the list. Then clear the bit and
```

```
121 0434 2 ! credit an entry to the LRU count.
122 0435 2 !
123 0436 2
124 0437 2 IF .FCB[FCB$W_ACNT] EQL 0
125 0438 2 THEN
126 0439 3 BEGIN
127 0440 3 IF NOT .FCB[FCB$V_DIR]
128 0441 3 THEN INSQUE (.FCB, .CURRENT_VCB[VCB$L_FCBBL])
129 0442 3 ELSE
130 0443 4 BEGIN
131 0444 4 FCB[FCB$V_DIR] = 0;
132 0445 4 CURRENT_VCB[VCB$B_LRU_LIM] = .CURRENT_VCB[VCB$B_LRU_LIM] + 1;
133 0446 3 END;
134 0447 2 END;
135 0448 2
136 0449 2 ! Now hook the window onto the FCB and adjust the access counts
137 0450 2 ! according to the access control bits in the window.
138 0451 2 !
139 0452 2
140 0453 2 WINDOW_SEGMENT = .WINDOW;
141 0454 2 DO
142 0455 3 BEGIN
143 0456 3 INSQUE (.WINDOW_SEGMENT, .FCB[FCB$L_WLBL]);
144 0457 3 WINDOW_SEGMENT = .WINDOW_SEGMENT[VCB$L_LINK];
145 0458 3 END
146 0459 2 UNTIL .WINDOW_SEGMENT EQL 0;
147 0460 2 FCB[FCB$W_ACNT] = .FCB[FCB$W_ACNT] + 1; ! bump access count
148 0461 2
149 0462 2 IF .WINDOW[VCB$V_NOREAD]
150 0463 2 THEN FCB[FCB$V_EXCL] = 1; ! set exclusive access
151 0464 2
152 0465 2 IF .WINDOW[VCB$V_NOWRITE]
153 0466 2 THEN FCB[FCB$W_LCNT] = .FCB[FCB$W_LCNT] + 1; ! no writers
154 0467 2
155 0468 2 IF .WINDOW[VCB$V_NOTRUNC]
156 0469 2 THEN FCB[FCB$W_TCNT] = .FCB[FCB$W_TCNT] + 1; ! no truncates
157 0470 2
158 0471 2 ! For a write access, bump the writer count. If this is the
159 0472 2 ! first write, and the file is the index file or the storage map, set
160 0473 2 ! the appropriate flag in the VCB.
161 0474 2 !
162 0475 2
163 0476 2 IF .WINDOW[VCB$V_WRITE]
164 0477 2 THEN
165 0478 3 BEGIN
166 0479 3 IF .FCB[FCB$W_WCNT] EQL 0
167 0480 3 THEN
168 0481 4 BEGIN
169 0482 4 IF .FCB[FCB$W_FID_NUM] EQL 1
170 0483 4 THEN CURRENT_VCB[VCB$V_WRITE_IF] = 1;
171 0484 4 IF .FCB[FCB$W_FID_NUM] EQL 2
172 0485 4 THEN CURRENT_VCB[VCB$V_WRITE_SM] = 1;
173 0486 3 END;
174 0487 3 FCB[FCB$W_WCNT] = .FCB[FCB$W_WCNT] + 1;
175 0488 2 END;
176 0489 2
177 0490 2 ! Count the access in the volume transaction count and return
```

```
178 0491 2 ! the window address for the user's CCB.
179 0492 2 !
180 0493 2
181 0494 2 PMSSGL_OPEN = .PMSSGL_OPEN + 1; ! bump open file count
182 0495 2 PMSSGL_OPENS = .PMSSGL_OPENS + 1; ! bump count of opens
183 0496 2 CURRENT_VCB[VCB$W_TRANS] = .CURRENT_VCB[VCB$W_TRANS] + 1;
184 0497 2
185 0498 2 ABD[ABD$C_WINDOW, ABD$W_COUNT] = 4; ! enable write-back
186 0499 2 .ABD[ABD$C_WINDOW, ABD$W_TEXT] + ABD[ABD$C_WINDOW, ABD$W_TEXT] + 1 = .WINDOW;
187 0500 2 ! put window address in buffer text
188 0501 2
189 0502 2 ! Mark the access complete in the cleanup action flags.
190 0503 2 !
191 0504 2
192 0505 2 CLEANUP_FLAGS[CLF_DEACCESS] = 1;
193 0506 2
194 0507 1 END; ! end of routine MAKE_ACCESS
```

```
.TITLE MAKACC
.IDENT \V04-000\
```

```
.EXTRN CLEANUP_FLAGS, CURRENT_VCB
.EXTRN PMSSGL_OPEN, PMSSGL_OPENS
```

```
.PSECT $CODE$,NOWRT,2
```

				0000G	CF	9E	00002		.ENTRY	MAKE_ACCESS, Save R2,R3	: 0377
		53			AC	D0	00007		MOVAB	CURRENT_VCB, R3	
		51		04	A1	B5	0000B		MOVL	FCB, R1	: 0437
				1A		12	0000E		TSTW	26(R1)	
					14				BNEQ	2\$	
		50			63	D0	00010		MOVL	CURRENT_VCB, R0	: 0441
		06		22	A1	E8	00013		BLBS	34(R1), 1\$	: 0440
	04	B0			61	0E	00017		INSQUE	(R1), @4(R0)	: 0441
					07	11	0001B		BRB	2\$	
	22	A1			01	8A	0001D	1\$:	BICB2	#1, 34(R1)	: 0444
				49	A0	96	00021		INCB	73(R0)	: 0445
		51		08	AC	D0	00024	2\$:	MOVL	WINDOW, WINDOW_SEGMENT	: 0453
		50		04	AC	D0	00028	3\$:	MOVL	FCB, R0	: 0456
	14	B0			61	0E	0002C		INSQUE	(WINDOW_SEGMENT), @20(R0)	
		51		20	A1	D0	00030		MOVL	32(WINDOW_SEGMENT), WINDOW_SEGMENT	: 0457
					F2	12	00034		BNEQ	3\$	: 0459
		50		04	AC	D0	00036		MOVL	FCB, R0	: 0460
				1A	A0	B6	0003A		INCB	26(R0)	
		52		08	AC	D0	0003D		MOVL	WINDOW, R2	: 0462
	04	15	A2		02	E1	00041		BBC	#2, 21(R2), 4\$	
		22	A0		08	88	00046		BISB2	#8, 34(R0)	: 0463
		03		14	A2	E9	0004A	4\$:	BLBC	20(R2), 5\$	: 0465
				1E	A0	B6	0004E		INCB	30(R0)	: 0466
	03	15	A2		03	E1	00051	5\$:	BBC	#3, 21(R2), 6\$	: 0468
				20	A0	B6	00056		INCB	32(R0)	: 0469
	22	0B	A2		01	E1	00059	6\$:	BBC	#1, 11(R2), 9\$	: 0476
				1C	A0	B5	0005E		TSTW	28(R0)	: 0479
					1A	12	00061		BNEQ	8\$	
		01		24	A0	B1	00063		CMPL	36(R0), #1	: 0482
					07	12	00067		BNEQ	7\$	

	51		63	D0	00069	MOVL	CURRENT_VCB, R1	: 0483
0B	A1		01	88	0006C	BISB2	#1, 11(R1)	: 0484
	02	24	A0	B1	00070	CMPW	36(R0), #2	: 0485
			07	12	00074	BNEQ	8\$	: 0487
0B	51		63	D0	00076	MOVL	CURRENT_VCB, R1	: 0494
	A1		02	88	00079	BISB2	#2, 11(R1)	: 0495
		1C	A0	B6	0007D	INCL	@#PMSS\$GL_OPEN	: 0496
		000000000G	9F	D6	00080	INCL	@#PMSS\$GL_OPENS	: 0498
		000000000G	9F	D6	00086	INCL	@#PMSS\$GL_OPENS	: 0499
	50		63	D0	0008C	MOVL	CURRENT_VCB, R0	: 0505
		0C	A0	B6	0008F	INCL	12(R0)	: 0507
	51	0C	AC	D0	00092	MOVL	ABD, R1	
02	A1		04	B0	00096	MOVW	#4, 2(R1)	
	50		61	3C	0009A	MOVZWL	(R1), R0	
		01	A140	9F	0009D	PUSHAB	1(R1)[R0]	
	9E		52	D0	000A1	MOVL	R2, @ (SP)+	
0000G	CF		01	88	000A4	BISB2	#1, CLEANUP_FLAGS+2	
			04	000A9	RET			

; Routine Size: 170 bytes,      Routine Base: \$CODE\$ + 0000

: 195            0508 1  
: 196            0509 1 END  
: 197            0510 0 ELUDOM

PSECT SUMMARY		
Name	Bytes	Attributes
\$CODE\$	170	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics					
File	-----		Symbols		Processing
	Total	Loaded	Percent	Pages Mapped	
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	23	0	1000	00:02.0

COMMAND QUALIFIERS

BI ISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:MAKACC/OBJ=OBJ\$:MAKACC MSRC\$:MAKACC/UPDATE=(ENH\$:MAKACC)

MAKACC  
V04-000

M 1  
16-Sep-1984 01:09:33

VAX-11 Bliss-32 V4.0-742

Page 7

; Size: 170 code + 0 data bytes  
; Run Time: 00:08.2  
; Elapsed Time: 00:26.4  
; Lines/CPU Min: 3754  
; Lexemes/CPU-Min: 16181  
; Memory Used: 114 pages  
; Compilation Complete

MAK  
VA)

Ph  
---  
Int  
Con  
Pas  
Syn  
Pas  
Syn  
Pse  
Cro  
Ass

The  
183  
The  
489  
13

Mac  
---  
\$2  
\$2  
TOT

278

The

MAC

0166 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

MAKACC  
LIS

MODIFY  
LIS

REQUEU  
LIS

RWATTR  
LIS

SCHFCB  
LIS

MPWIND  
LIS

MAPUBN  
LIS

PMS  
LIS

RDHEDR  
LIS

RWUB  
LIS

RETDIR  
LIS

ROBLOK  
LIS

SMALOC  
LIS

MOUNT  
LIS

MAKMBE  
LIS

MAKSTR  
LIS

MAKTHOR  
LIS